

Rapport de projet annuel

Abstraction de vidéos

Abdelkader BENCHIKH

Jury :

Yves Lepage

Jacques Madelene

Tuteur universitaire :

Youssef Chahir

Dans le cadre de ma deuxième année du Master informatique RADIS à l'université de CAEN Basse-Normandie, et afin d'appliquer les méthodologies et les notions enseignées je dois réaliser un projet, celui-ci permet d'appliquer les connaissances acquises durant ma scolarité. Le projet que je dois réaliser consiste à implémenter un algorithme de transformation d'une image à un dessin, qui consiste à appliquer un filtre particulier sur l'image en respectant certaines contraintes de voisinage et en essayant d'amplifier les contrastes entre les différentes zones de l'image. L'objectif final est de « convertir » une séquence d'images animées vidéo en un dessin animé.

Afin de comprendre la démarche que j'ai utilisée pour mener ce projet à son terme, mon rapport se structure de la façon suivante :

Tout d'abord, dans une première partie je présente quelques concepts de base dans l'imagerie et le cadre général de mon projet. Puis dans une seconde partie, je présente l'algorithme de base de transformation d'une image à un dessin. Ensuite dans une troisième partie, je explique comment j'ai développé l'outil précédent pour transformer une vidéo en dessin animé, avant que dans une quatrième partie je décrive l'interface qui permet de utiliser l'outil de transformation.

Table des matières

1.Introduction :	5
2.Présentation des concepts de base :	6
2.1.Espace de couleurs :	6
2.2.Le filtrage :	8
2.3.Détection des contours:	9
3.Transformation d'une image :	10
3.1. Algorithme :	10
i.Conversion de RGB vers LAB :	13
ii.Filtre bilatéral :	14
iii.Dog Edge :	17
iv.Filtre bilatéral et Dog Edge :	18
v.La quantification :	19
3.2. Exemples :	20
4.Transformation d'une vidéo :	21
5.Environment de travail :	22
6.Interface :	23
7.Conclusion :	28

1. Introduction :

L'abstraction ou la stylisation d'une image est un moyen efficace pour la communication visuelle, elle permet de simplifier l'affichage, et attirer l'attention vers le plus significatif. L'abstraction d'une image consiste à supprimer les détails moins importants et accroître les pertinences détails, ce qui veut dire, préserver et souligner les éléments clés de la forme visuelle, et identifier le contenu significatif de l'image et aucune talent artistique est nécessaire, on cherche tout simplement les information pertinentes dans l'image grâce à une série des opérations, et on donne des couleurs constantes aux régions en préservant la structure de l'image.

Le bute de ce projet de concevoir un outil qui permet dans un premier temps de construire un dessin à partir d'une image, et de construire un dessin animé à partir d'une vidéo dans un seconde temps.

J'ai basé sur l'étude effectuée par des étudiants de l'université North Western qui définit un algorithme de base de transformation.

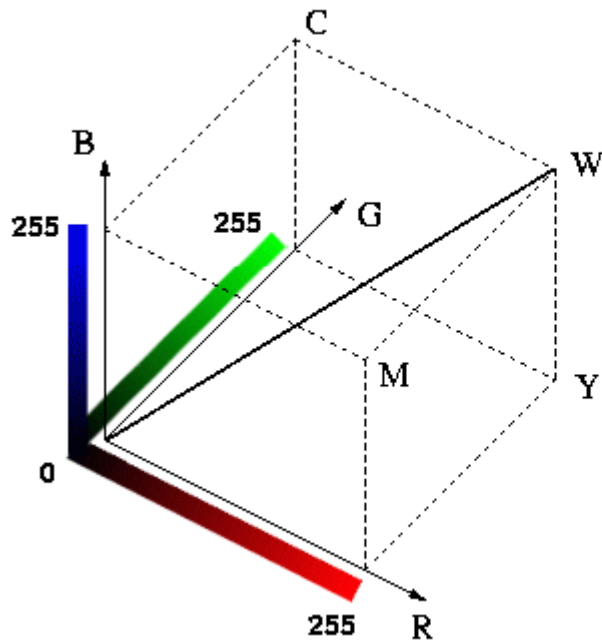
2. Présentation des concepts de base :

2.1. Espace de couleurs :

Espace de couleur ou espace colorimétrique représente une couleur par une combinaison numérique, en générale ces combinaisons sont définies sous forme de triplets, donc chaque couleur est représentée par un point dans un espace de trois dimensions.

Il y plusieurs espaces de couleurs :

- **RGB** : dans cet espace une couleur est caractérisée par un pixel de trois valeurs (valeurs entières) qui représentent le niveau des trois composantes Rouge Vert Bleu de cette couleur.



R: red
G: green
B: blue
C: cyan
M: magenta
Y: yellow
W: white

- **LAB** : dans cet espace une couleur est caractérisée par trois paramètres : une valeur d'intensité correspondant à la luminance et les deux autres paramètres de chrominance qui décrivent la couleur.

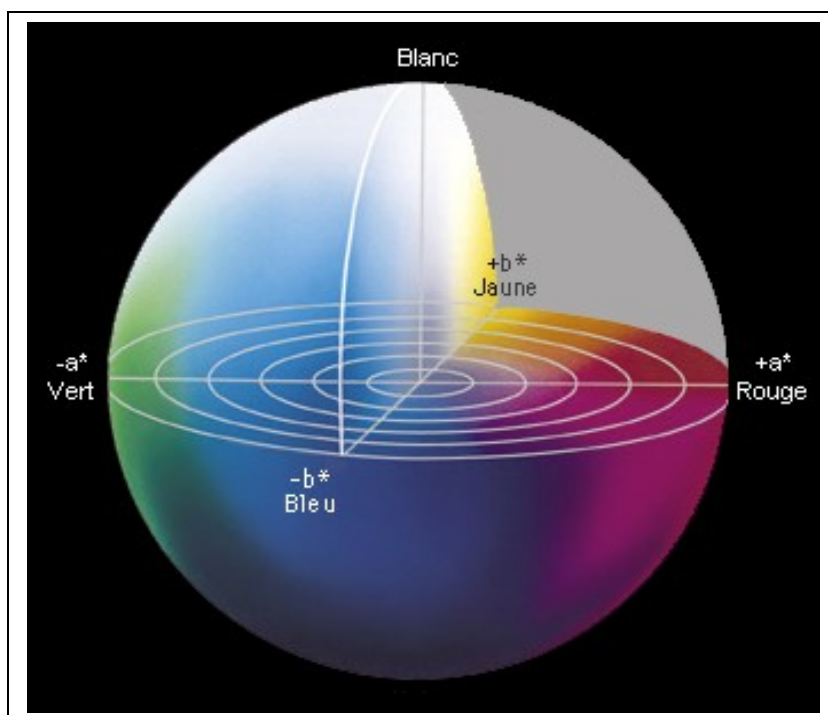
Cet espace de couleur a trois axes principales :

L : la clarté :qui va de noir(0) a blanc (100)

A : représente la gamme de l'axe : rouge vers vert

B : représente la gamme de l'axe jaune vers bleu

Voici son modèle démonstratif :



Comme il y a plusieurs d'autres espaces de couleurs : CIE, YIQ, YUV, TSL...

2.2.Le filtrage :

Un filtre c'est une transformation mathématique permet d'améliorer la qualité de l'image ou d'en extraire de l'information. Une fois le filtre est appliqué sur toute ou une partie de l'image, la valeur de chaque pixel de cette dernière est modifiée en fonction des valeurs des pixels voisins.

Le filtrage permet de réaliser une opération fondamentale dans l'imagerie c'est *le lissage*, cette opération consiste à *atténuer ou supprimer les aspérités et petits détails dans une image.* .

Il y a deux catégories de filtres :

Filtre linéaire : la valeur du nouveau pixel dépend du voisinage, et ne pas de la position dans l'image, exemple :

- *Filtre uniforme.*
- *Filtre pyramidal.*
- *Filtre gaussien.*

Filtre non-linéaire : est plus complexe à mettre en œuvre que le filtre linéaire, la valeur du nouveau pixel dépend du voisinage et la position dans l'image, ce qui permet de donner des résultats de qualité meilleur et de préserver les discontinuités, exemple :

- *Filtre médian.*
 - *Filtre bilatéral.*
-

2.3.Détection des contours:

La détection des contours est une étape importante dans l'analyse d'images, elle consiste à repérer les pixels où il y a un changement brutal de la valeur, ce qui correspond aux bords entre les différentes régions de l'image.

La détection des contours revient à la découpage de l'image en régions homogènes, en général les régions homogènes sont des régions où l'intensité lumineuse varie lentement en fonction des coordonnées spatiales. Les contours sont des portions de l'image, de faible largeur séparent les régions dans une image.

Il y a deux approches pour la détection des contours :

- Approche gradient
- Approche Laplacien

Ces approches reposent sur le fait que les contours correspondent des discontinuités d'ordre 0 de la fonction d'intensité

3. Transformation d'une image :

Un dessin est un simple moyen de communication visuelle, qui fait l'interprétation d'une scène réelle, dans un dessin on utilise une quantité minimale d'information et représente efficacement un objet.

La construction d'un dessin à partir d'une image consiste à augmenter le contraste des régions, faire perdre les détails qui ne sont pas importants et utiliser des couleurs douces de quantification pour créer un effet de type cartonné.

On peut citer deux différences principales entre une image et son dessin équivalent :

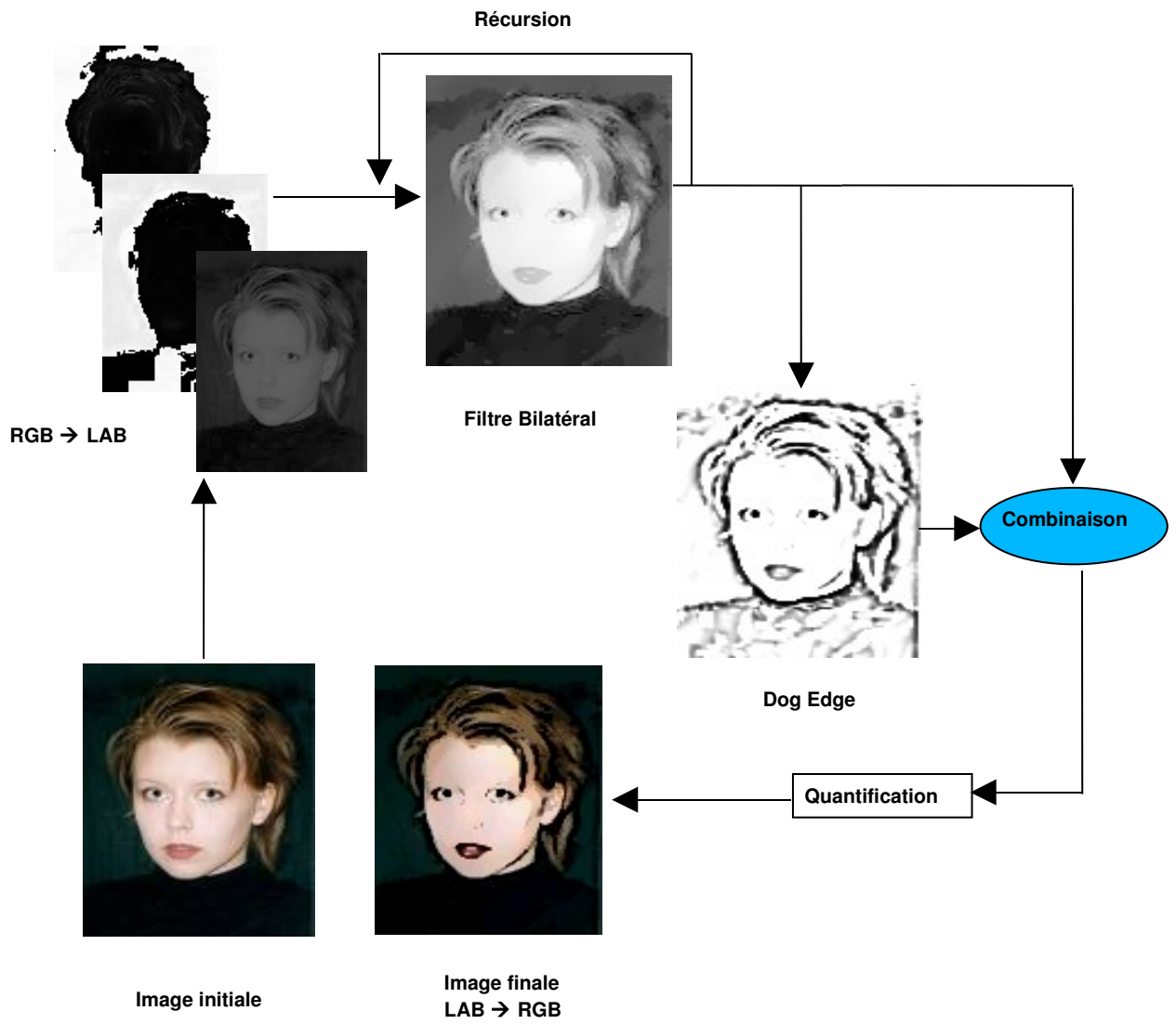
- 1- la texture : dans le dessin on a moins de détails que l'image originale
- 2- le contraste : dans le dessin le contraste est plus important que dans l'image

3.1. Algorithme :

L'algorithme de transformation se constitue de plusieurs étapes :

- Conversion de l'espace de couleur : dans cette étape on convertit l'espace de couleur de l'image initiale (RGB) vers l'espace de couleurs (LAB) pour travailler sur l'image au niveau de gris dans le reste de l'algorithme
- Filtrage : pour but de faire un lissage et faire disparaître les détails les moins importants
- Détection de contours : on sélectionne les bords entre les régions de l'image, et on les amplifie grâce à l'approche de DOG Edge
- Combinaison : c'est la combinaison entre l'image lissée et l'image des contours pour avoir une image qui ressemble un dessin
- Quantification : cette étape n'est pas primordiale, elle permet de réduire le nombre de couleurs pour améliorer la qualité du dessin finale
- Conversion inverse d'espace de couleurs : la conversion de LAB vers RGB

Le schéma suivant illustre l'enchaînement des étapes de l'algorithme de transformation :



i. Conversion de RGB vers LAB :

L'espace de couleur RGB(Rouge,Vert,Bleu) représente chaque couleur de l'image par trois valeurs correspondent au niveau de ses composants RVB, et pour but de travailler sur une image au niveau de gris, on fait une conversion vers un autre espace de couleur c'est le LAB.

Cette conversion permet d'avoir trois images correspondent au composante LAB comme il montre l'exemple ci-dessous :

Image initiale



Composant L



Composant A



Composant B

L'obtention d'une image en luminance (en niveau de gris) où chaque couleur est représentée par une seule valeur, ça nous facilite les calculs, comme on a la possibilité de conserver l'homogénéité des couleurs pour pouvoir récupérer l'image initiale en faisant la conversion inverse LAB vers RGB dans la dernière étape de l'algorithme

ii. Filtre bilatéral :

C'est un filtre non- linéaire qui base sur des gaussiennes spatiale et intensité, permet de faire un lissage et éliminer des détails qui ne sont pas utiles et , avec l'avantage de préserver les contours entre les régions de l'image, c'est-à-dire quand on fait la lissage on ne se déplace que dans la zones semblable.

La formule mathématique de filtre est la suivante :

$$U(\hat{x}, \hat{y}) = \frac{\int_{-3\sigma}^{3\sigma} W_r \cdot W_s \cdot U(\hat{x}+i, \hat{y}+j) \cdot d(d_{ij})}{\int_{-3\sigma}^{3\sigma} W_r \cdot W_s \cdot d(d_{ij})}$$

- U(x,y) : la valeur du pixel
- Wr : fonction d'éloignement en intensité
- Ws : fonction d'éloignement spatial
- d : la distance
- σ :

▪ **Eloignement spatiale :**

La fonction d'éloignement spatiale, permet de lisser les faibles discontinuités, c'est-à-dire si les pixels sont proches en intensité on peut les moyenner.

$$W_s = e^{-\frac{1}{2} (d_{ij} / \sigma_s)^2}$$

- dij :la distance entre le pixels ses pixels voisins, $d_{ij}^2 = x*x + y*y$
- σs :sigma spatiale

Pour chaque pixel on prends ses pixels voisins, et on fait un moyennage pour lisser une partie de l'image, le paramètre sigma détermine la taille de la surface à moyenner, si on prend un sigma plus grand on prendra plus de pixels voisins, ce qui influence beaucoup sur le temps de calcul. On prendra une faible valeur de sigma pour réduire le temps d'exécution .

Voici les résultats obtenus avec des valeurs différentes de sigma spatial :



$\sigma_s=1$



$\sigma_s=2$



$\sigma_s=3$



$\sigma_s=5$

- **Eloignement en intensité :**

Pour lisser l'image en gardant les fortes discontinuités grâce au sigma intensité :

$$W_r = e^{-\frac{1}{2} \left[\frac{U(\hat{x}, \hat{y}) - U(\hat{x}+i, \hat{y}+j)}{\sigma_{int}} \right]^2}$$

σ_{int} : sigma intensité

Voici les résultats obtenus en changeant la valeur de ce paramètre :



$\sigma_{int}=0.5$



$\sigma_{int}=1$



$\sigma_{int}=3$



$\sigma_{int}=5$

Ces résultats montrent si on prend un sigma intensité plus grand, ça floute beaucoup l'image sans préserver les contours, donc est fortement conseillé de prendre une valeur faible, et pour avoir plus d'effet sans détruire les discontinuités on peut réitérer tout le filtre.

Exemple:



2 itérations, $\sigma_{int}=1.25$



4 itérations, $\sigma_{int}=1.25$



7 itérations, $\sigma_{int}=1.25$

iii. Dog Edge :

Cette étape est pour bute d'extraire les lignes de style qui donne la forme de l'image, ces lignes (les bords) sont définis par une haute contraste en générale, donc afin d'ajouter visuellement des bords de haut niveau de contraste à des régions distinctes, on met en œuvre une approche qui base sur la différence de gaussienne (Dog).

$$D(\hat{x}, \sigma_e, \tau, \varphi_e) = \begin{cases} Si & (S(\sigma_e) - \tau.S(\sigma_r) > 0) & 1 \\ Sinon & 1 + \tan(\varphi_e.(S(\sigma_e) - \tau.S(\sigma_r))) \end{cases}$$

L'idée de cette approche :

Pour détecter les contours d'une image, on applique deux fonction de floue(blure) sur cette dernière, en suite on fait la différence (différence de gaussienne)entre les deux images obtenues,si cette différence est positive ça signifie qu'il n'y a pas de contraste importante entre les régions de l'image, sinon il y a une contraste importante il faut l'amplifier.

Dans la formule plusieurs paramètres interviennent :

S : c'est la fonction de floue, qui mélange chaque pixel avec plus ou moins ses pixels voisins

φ_e :correspond à l'intensité des contours

τ :c'est le paramètre qui détermine le niveau de sensibilité de la détection, si pi est petit donc c'est moins de bruit détecté ce qui correspond à non présence du bord important et vis vers ça.

σ_e :correspond à l'épaisseur du trait noir du contour.

σ_r :une valeur typique($\sigma_r = \text{racine } \sigma_e * 1.6$)

 $\tau=0.998$  $\tau=0.9$  $\tau=1$

iv. Filtre bilatéral et Dog Edge :

Les deux étapes représentent une partie très importante dans l'algorithme de transformation montrée précédemment, mais il faut quand on sache que le filtre fait un lissage de l'image en moyennant la valeur d'un pixel avec ses pixels voisins, ce qui en fait va éteindre l'importance de la contraste entre les régions, et vu que le Dog edge fait la détection des contours en basant sur la comparaison de la luminance des pixels, donc le fait de faire trop de lissage ça peut poser un problème et diminuer la détection, donc ce quand essaierai de faire est de ne pas faire trop de lissage avant de faire passer l'opération de Dog Edge, une fois cette dernière est fait on peut faire le lissage autant quand veut

Voici quelques exemples qui illustrent la relation entre le filtre bilatéral et le Dog Edge :



2 itérations puis Dog puis
2 itérations



4 itérations puis Dog

v. La quantification :

La quantification c'est l'opération qui permet d'avoir une image avec une plus petite palette de couleurs par rapport à l'image originale, en minimisant la différence visuelle.

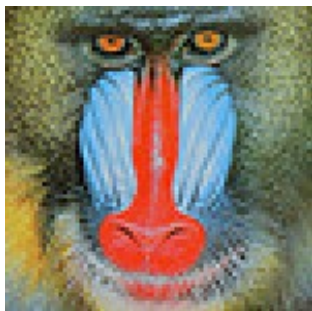
Cette étape dans notre algorithme n'est pas très importante, c'est juste pour améliorer la qualité du dessin final.

J'ai réalisé plusieurs essais en gardant les mêmes paramètres et en changeant le nombre de couleurs :

Image initiale :



3.2. Exemples :



4. Transformation d'une vidéo :

Vidéos :

Dans la première partie de ce projet, on a réussi à réaliser un outil qui permet d'obtenir un dessin à partir d'une image, et on s'aperçoit que les résultats sont bons. ce qu'on fait dans cette partie, est de construire un dessin animé à partir d'une vidéo.

On va pas se contenter de découper la vidéos en séries de frames, les transformer en dessins et les rassembler pour obtenir une vidéos, mais plutôt considérer la vidéo comme une image à trois dimensions, ce qui permet d'avoir plus d'informations en prenant plus de voisinage, et ce qui pourra donner une qualité meilleur.

Avec des valeurs typiques données aux paramètres qui interviennent dans la construction du dessin on peut avoir des bons résultats, mais on a la possibilité de modifier ces valeurs pour faire des améliorations.

L'algorithme de construction du dessin est réutilisé en rajoutant la 3ème dimension, et de choisir des valeurs qui permettent de bien faire apparaître les contours, et faire des lissages pour donner l'effet d'un dessin.

La transformation d'une vidéo est très coûteuse au niveau du temps, donc c'est important de prendre les bons paramètres qui minimisent les calculs.

Prendre des paramètres standards ce n'est pas la meilleurs solution, mais selon la seine de la vidéo on peut jouer avec ces paramètres pour avoir une qualité meilleur du dessin animé.

5. Environnement de travail :

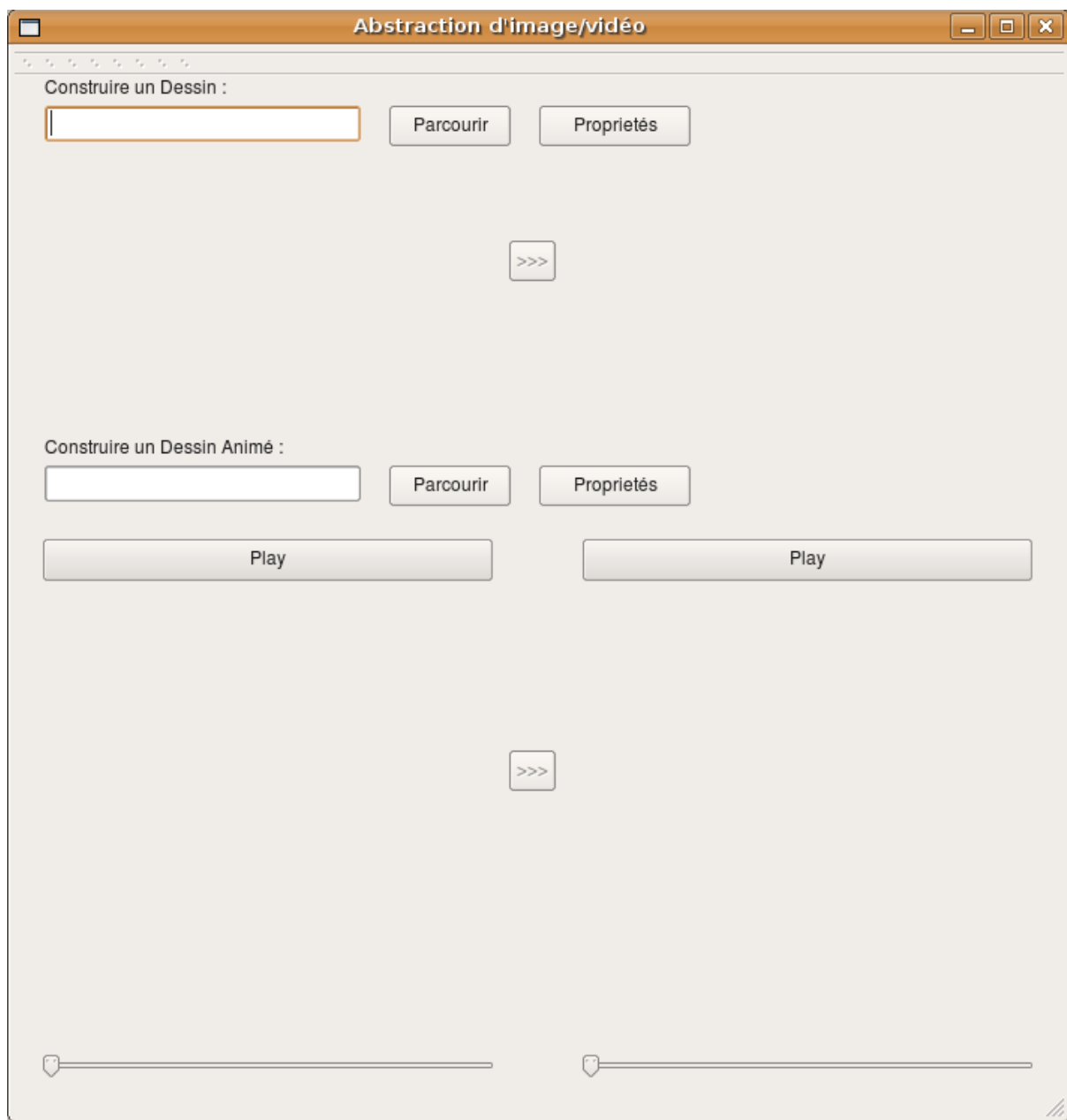
J'ai choisi le langage de programmation orienté objet C++ pour implémenter l'algorithme de transformation, dans l'environnement de développement Intégré NetBeans avec la version 6.4

Pour faciliter le traitement de l'image j'ai utilisé la librairie CImg(2.128) qui fournit des plusieurs fonctionnalités qui sont très utiles dans mon projet .

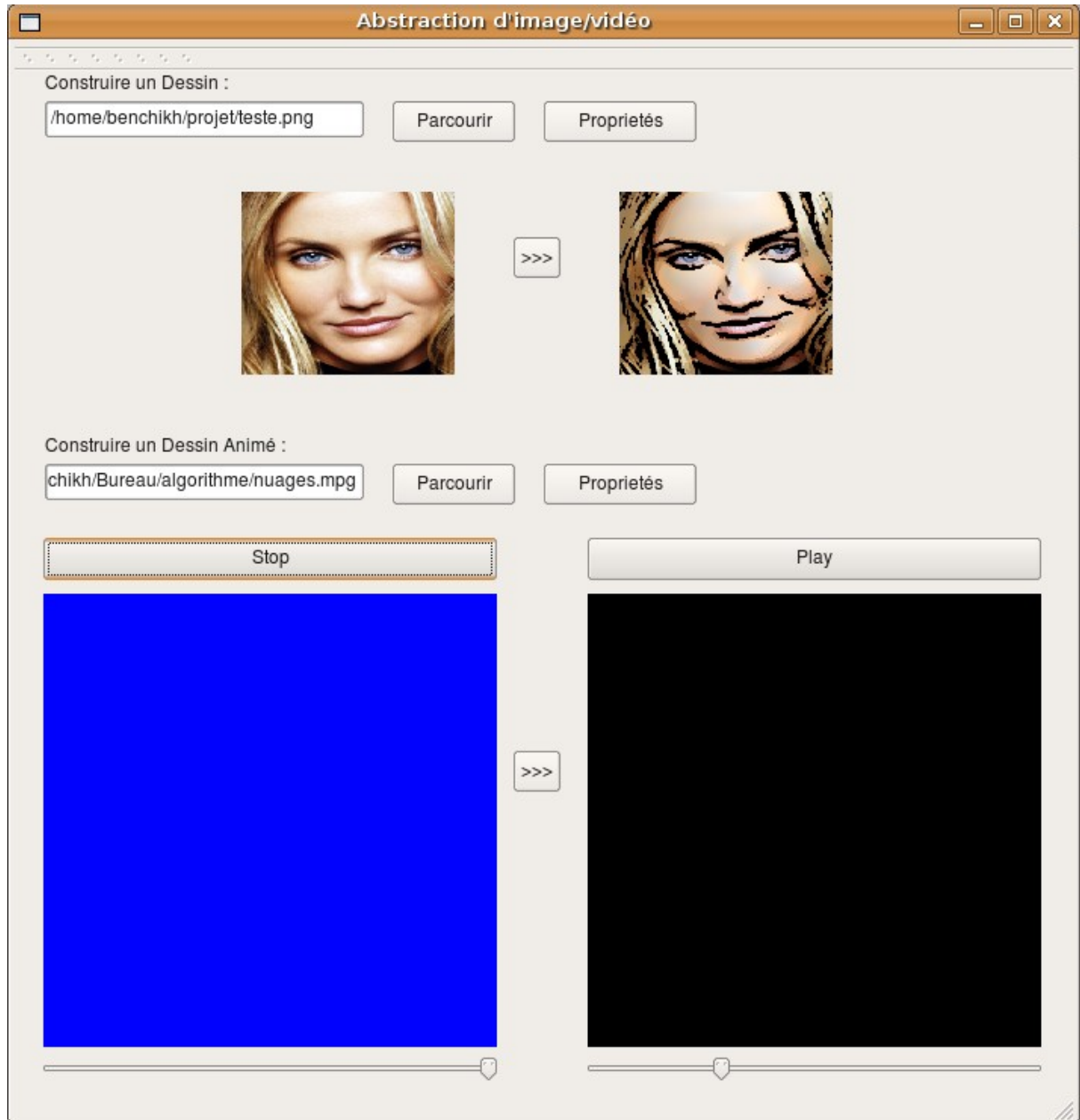
Pour réaliser l'interface qui permet aux utilisateurs de faire la transformation d'une image ou d'une vidéo, j'ai utilisé la librairie Qt4

6. Interface :

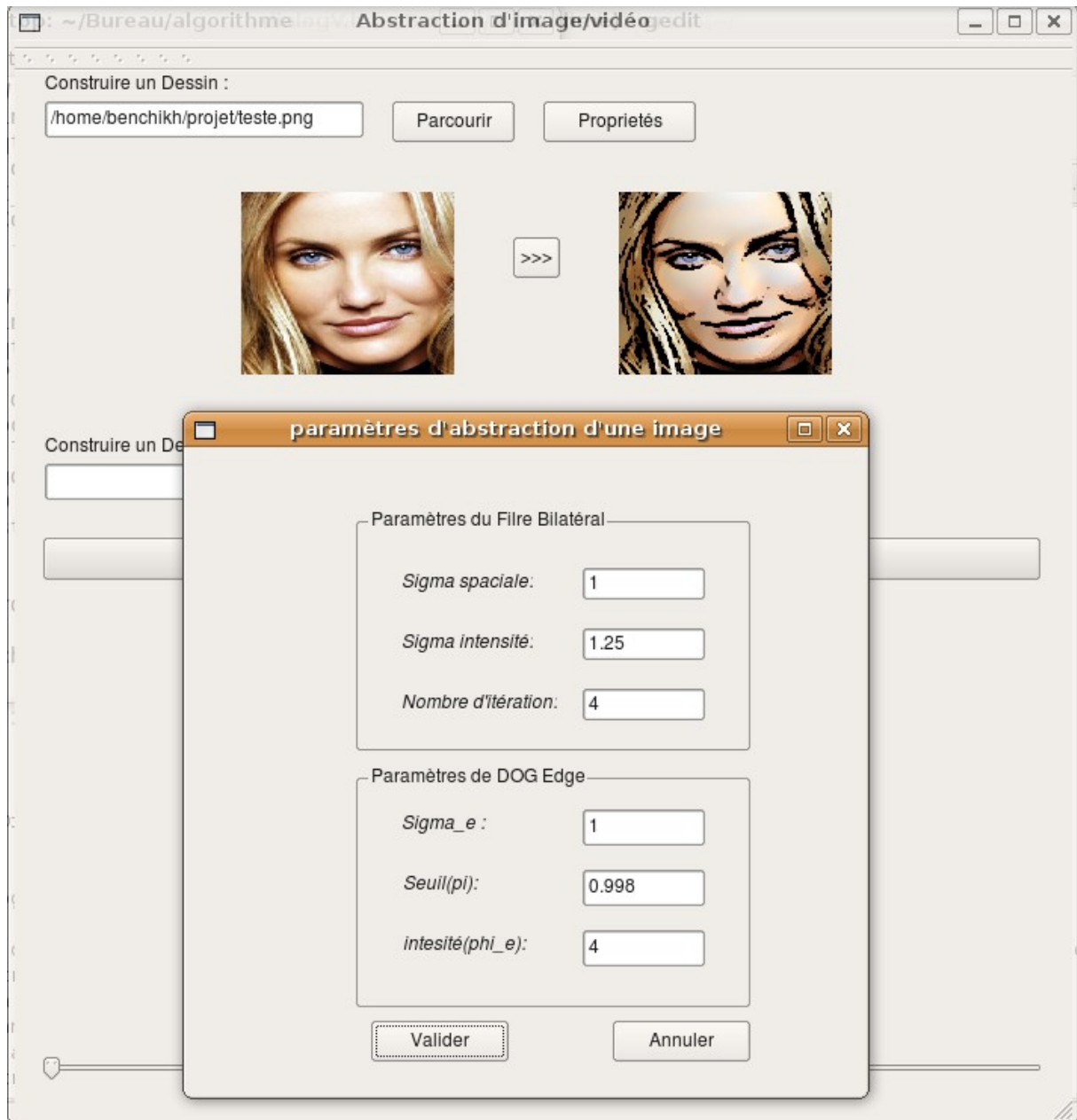
Pour faciliter l'utilisation de cet outil, j'ai réalisé une interface qui permet de choisir l'image ou la vidéo (bouton parcourir), choisir les valeurs des paramètres (bouton propriétés) et faire la transformation en cliquant sur un bouton approprié (>>>).



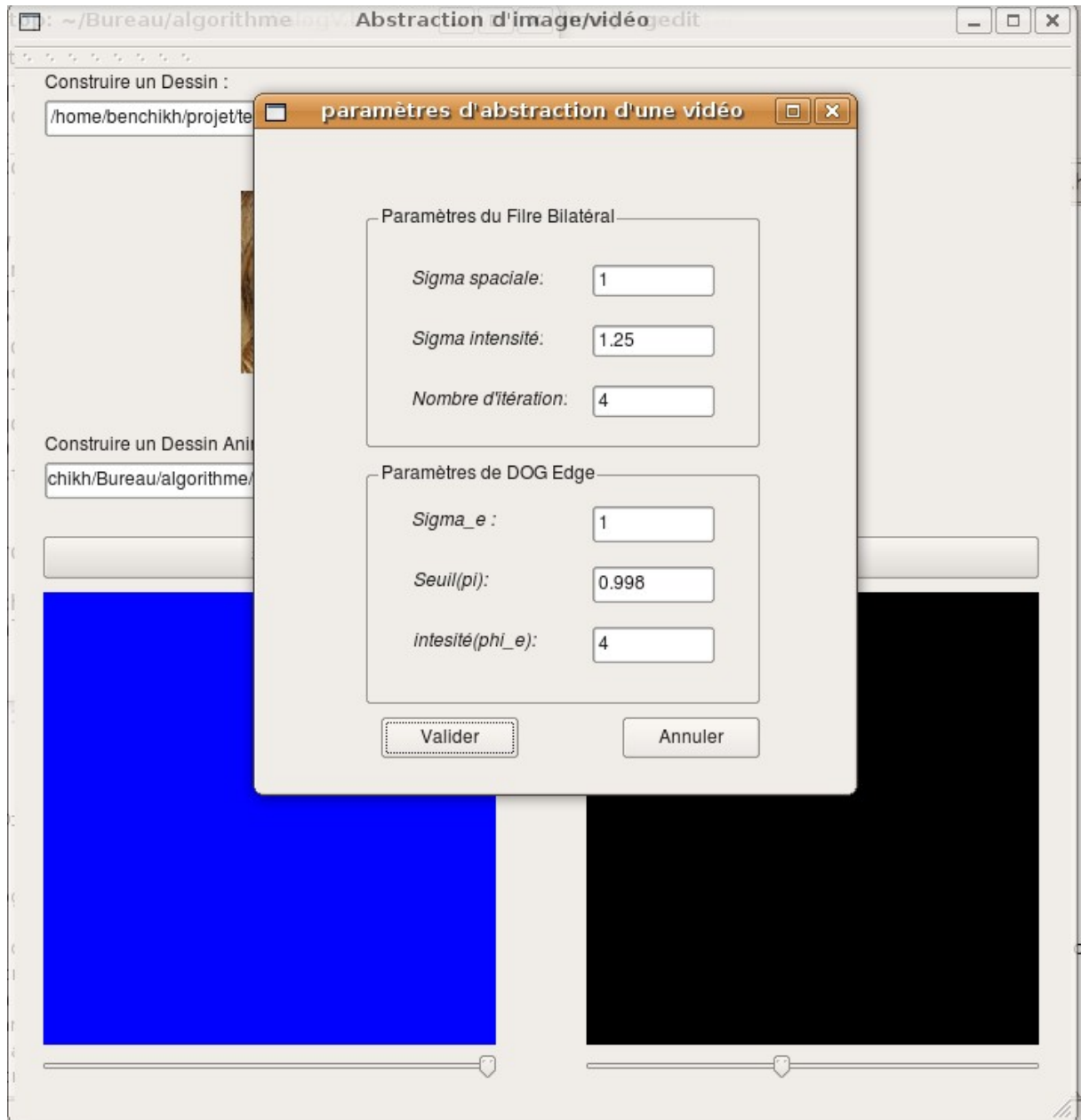
Quant on clique sur le bouton de transformation, un appel de la fonction de construction du dessin ou dessin animée est fais.



Quand on clique sur le bouton propriétés, ça va ouvrir une fenêtre avec les paramètres par défaut, et c'est à l'utilisateur de choisir les paramètres pour améliorer la qualité des résultats.



Pour le paramétrage de la vidéo il suffit de cliquer sur le bouton propriétés d'une vidéo :



7. Conclusion :

Pour conclure, le projet m'a permis d'approcher de près les composants importants de l'imagerie que ce soit le filtrage, la détection des contours, les espaces de couleurs.

Il y avait deux phases d'implémentation dans ce projet, les premiers mois je suis intéressé par la transformation d'une image à un dessin, en implémentant un algorithme de transformation, ce qui donnait des résultats acceptable avec des paramètres typique, et qui réduire le temps d'exécution de l'algorithme.

Dans la deuxième phase du projet, je travaillé sur la vidéo qui considérée comme une image sur trois plan, ce qui permet d'avoir des informations en plus et donner une qualité meilleur du résultats.

Dans un dernier temps, j'ai réalisé une interface qui permet de modifier les paramètres de la transformation, qui peut faciliter le paramétrage pour la vidéo malgré que les valeurs par défaut donnent des bons résultats.

Référence :

- Stylization and Abstraction of Photographs: Department of
Computer Science & Center for Cognitive Science
Rutgers University
